REMARKS

This communication is intended as a full and complete response to the Office Action issued May 21, 2007. In view of the following discussion, the Applicants submit that none of the claims now pending in the application are directed to non-statutory subject matter under the provisions of 35 U.S.C. §101 or obvious under the provisions of 35 U.S.C. §103. Thus, the Applicants believe that all of these claims are in allowable form.

I. REJECTION OF CLAIMS 14 AND 16-20 UNDER 35 U.S.C. §101

The Examiner objects to claims 14 and 16-20 under 35 U.S.C. §101, for being allegedly directed to non-statutory subject matter. In response, the Applicants have amended claims 14 and 16-20 in order to more clearly recite aspects of the invention.

Specifically, claims 14 and 16-20 have been amended, as suggested by the Examiner, to recite "computer-readable media for storing software instructions" that "when executed by a processor", cause the processor to execute a plurality of recited steps. This replaces the recitation of a "computer program product" comprising a plurality of computer codes for performing steps.

Thus, claims 14 and 16-20, as amended recite computer-readable media for storing software instructions that, when executed by a processor, cause a packet of data in a client/server object-based computing system to be transmitted. As the Supreme Court has recognized, Congress chose the expansive language of 35 USC §101 so as to include "anything under the sun that is made by man" as statutory subject matter. *Diamond v. Chakrabarty*, 447 U.S. 303, 308-09, 206 USPQ 193, 197 (1980) (MPEP 2106, emphasis added). The Applicants respectfully submit that computer-readable media carrying instructions that are readable by a processor, are made by man and are not naturally occurring phenomena.

Moreover, the Examiner submits that the text of 35 U.S.C. 101 requires patentable subject matter to fall into one of four enumerated categories, namely, "process, machine, manufacture or composition of matter" (See, Office Action, Page 3, emphasis added). The Applicants submit that a computer-readable medium carrying instructions that are readable by a processor, qualifies as a "manufacture" under the provisions of 35 U.S.C. §101, which allows for a broad interpretation of the term

"manufacture" (*See, e.g.*, MPEP 2105). For instance, the Court in *Chakrabarty* found that "in choosing such expansive terms as 'manufacture' and 'composition of matter,' modified by the comprehensive 'any,' Congress plainly contemplated that the patent laws would be given wide scope". The Court further found that a "nonnaturally occurring manufacture or composition of matter - a product of human ingenuity -having a distinctive name, character, [and] use" is patentable subject matter (emphasis added). As discussed above, computer readable media that carries instructions readable by a processor is not a naturally occurring phenomena, but rather owes existence to human ingenuity and intervention that bestows "new forms, qualities, properties, or combinations". Thus, the Applicants respectfully submit that the invention recited in claims 14 and 16-20 is at least a "manufacture" within the meaning of 35 U.S.C. §101.

In addition, the Examiner submits that claims 14 and 16-20 "fail[] to produce useful, concrete and tangible results". The Applicants respectfully disagree.

MPEP 2106 states that a "useful" invention is one for which the utility is "(i) specific, (ii) substantial and (iii) credible". In this case, the Applicant submits that the utility of the claimed invention is specific (the transmission of data packets), substantial (aids in the maintenance of reliable data distribution in client-server object-based computing systems), and credible (transmitting updates about objects in the form of data packets will improve the reliability of data distribution).

A "concrete" result, according to MPEP 2106, is one that is "substantially repeatable". The transmission of a data packet from a first computing device to a second computing device is repeatable and predictable.

A "tangible" result, according to MPEP 2106, is one that produces a "real-world result". As stated above, the result of the claimed invention is the reliable distribution of data in a client/server object-based computing system. This distribution is clearly an observable, "real-word" result.

Accordingly, the Applicants respectfully submit that the subject matter to which claims 14 and 16-20 is drawn is patentable, and, as such respectfully request that the rejection of claims 14 and 16-20 under U.S.C. §101 be withdrawn.

II. REJECTION OF CLAIMS 1, 3-6, 9, 10, 12-14, 16-20 AND 34 UNDER 35 U.S.C. §103

Claims 1, 3-6, 9, 10, 12-14, 16-20 stand rejected as being made obvious by the Lathrop patent (United States Patent No. 5,701,427, issued December 23, 1997, hereinafter "Lathrop") in view of the Wesley patent (United States Patent No. 6,076,114, issued June 13, 2000, hereinafter "Wesley") and further in view of the Ma et al. patent (United States Patent No. 5,920,725, issued July 6, 1999, hereinafter "Ma"). In response, the Applicants have amended independent claims 1, 9, 14, and 18, from which claims 3-5, 10, 12-13, 16-17, and 19-20 depend, as well as independent claim 34, in order to more clearly recite aspects of the present invention. Claim 6 has been cancelled without prejudice.

In particular, the Applicants respectfully submit that Lathrop, Wesley, and Ma, individually or in any permissible combination, fail to teach, show, or suggest the novel invention of, in a client/server object-based computing system including a first computing device and a second computing device, attempting to send a data packet representing an object in which the second computing device has interest by placing the packet of data in a queue of objects in which the second computing device has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue, as claimed by the Applicants' independent claims 1, 9, 14,18, and 34.

By contrast, the combination of cited references at most teaches that prioritized transmissions may influence the latency experienced by a sender on a communication link (*See, e.g.*, Wesley at column 8, lines 14-18: "an increase in latency may be due to ...prioritized transmissions incorporated in various wireless network routing protocols ..."). However, the references do not suggest how to implement such prioritized transmissions, and certainly do not suggest that such prioritized transmissions may be implemented by use of a queue.

Moreover, even if the prioritized transmissions mentioned by Wesley could be considered to suggest prioritizing packets maintained in a queue, the references do not teach that the queue (and, presumably, its associated prioritization scheme) is maintained by the sender of the packet (*e.g.*, a server). At best, Wesley suggests that

transmissions are prioritized by the receiver (*See, e.g.*, Wesley at column 8, lines 18-22: "the receiving station on the communications link may have given a higher priority to other data packets", emphasis added).

Notably, Applicants' claims positively recite the step of attempting to send, in a client/server object-based computing system including a first computing device and a second computing device, a data packet representing an object in which the second computing device has interest by placing the packet of data in a queue of objects in which the second computing device has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue. Specifically, Applicants' independent claims 1, 9, 14, 18 and 34 recite:

> 1. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:
>
> identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being identified as an object for which the second computing system has an interest in receiving updates, the object including data and functionality;
>
> attempting to send the packet of data from the first computing system to the second computing system, wherein attempting to send the packet of data from the first computing system to the second computing system includes:
>
> placing the packet of data in a queue of objects in which the second computing system has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue; and
>
> removing the packet of data from the queue using the second computing system;
>
> determining when the packet of data is received by the second computing system;
>
> sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system; and
>
> re-attempting at least once to send the packet of data from the first computing system to the second computing system when it is determined that

the packet of data is not received by the second computing system, wherein a time differential between each re-attempt is determined using statistical information including at least one measurement of an amount of time elapsed for another packet of data to be sent and received. (Emphasis added)

9. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:

a) attempting to send the packet of data from the first computing system to the second computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being identified as an object for which the second computing system has an interest in receiving updates, the object including data and functionality, and wherein attempting to send the packet of data from the first computing system to the second computing system includes:

placing the packet of data in a queue of objects in which the second computing system has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue; and

removing the packet of data from the queue using the second computing system;;

b) determining when the packet of data is received by the second computing system;

c) identifying the packet of data as being successfully sent when it is determined that the packet of data is received by the second computing system; and

d) assuming that packet losses have occurred when it is determined that the packet of data is not received by the second computing system, wherein assuming that packet losses have occurred includes repeating a) and b) for up to a predetermined maximum number of times, wherein a time differential between each attempt at repeating a) and b) is determined using statistical information including at least one measurement of an amount of time elapsed for another packet of data to be sent and received. (Emphasis added)

14. A computer-readable media for storing software instructions for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, such that the instructions, when executed by a processor, perform the steps of:

computer code for identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being identified as an object for which the second computing system has an interest in receiving updates, the object including data and functionality;

computer code for attempting to send the packet of data from the first computing system to the second computing system, wherein attempting to send the packet of data from the first computing system to the second computing system includes:

placing the packet of data in a queue of objects in which the second computing system has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue; and

removing the packet of data from the queue using the second computing system;;

computer code for determining when the packet of data is received by the second computing system;

computer code for sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system;

computer code for re-attempting at least once to send the packet of data from the first computing system to the second computing system when it is determined that the packet of data is not received by the second computing system, wherein a time differential between each re-attempt is determined using statistical information including at least one measurement of an amount of time elapsed for another packet of data to be sent and received; and

a computer readable medium that stores the computer codes. (Emphasis added)

18. A computer-readable media for storing software instructions for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, such that the instructions, when executed by a processor, perform the steps of:

computer code for attempting to send the packet of data from the first computing system to the second computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being identified as an object for which the second computing system has an interest in receiving updates, the object including data and functionality, and

wherein attempting to send the packet of data from the first computing system to the second computing system includes:

placing the packet of data in a queue of objects in which the second computing system has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue; and

removing the packet of data from the queue using the second computing system;

computer code for determining when the packet of data is received by the second computing system;

computer code for identifying the packet of data as being successfully sent when it is determined that the packet of data is received by the second computing system;

computer code for assuming that packet losses have occurred when it is determined that the packet of data is not received by the second computing system, wherein assuming that packet losses have occurred includes computer code for re-attempting to send the packet of data from the first computing system to the second computing system and computer code for determining when the re-attempt to send the packet of data is successful for up to a predetermined maximum number of times, wherein a time differential between each re-attempt is determined using statistical information including at least one measurement of an amount of time elapsed for another packet of data to be sent and received; and

a computer readable medium that stores the computer codes. (Emphasis added)

34. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:

identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being represented in an object list in the first computing system, the object list arranged to include objects that are to be updated, and the object also being represented in a filter tree which is arranged to identify objects that the second computing system has an interest in, the object including data and functionality;

attempting to send the packet of data from the first computing system to the second computing system, wherein attempting to send the packet of data from the first computing system to the second computing system includes:

placing the packet of data in a queue of objects in which the second computing system has interest using the first computing system, the queue being maintained by the first computing system and arranged to

> prioritize the packet of data with respect to any other packets of data included in the queue; and
>
> removing the packet of data from the queue using the second computing system;
>
> determining when the packet of data is received by the second computing system; and
>
> sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system; and
>
> re-attempting at least once to send the packet of data from the first computing system to the second computing system when it is determined that the packet of data is not received by the second computing system, wherein a time differential between each re-attempt is determined using statistical information including at least one measurement of an amount of time elapsed for another packet of data to be sent and received. (Emphasis added)

As discussed above, Lathrop, Wesley and Ma, singly and in any permissible combination, fail to teach or suggest attempting to send, in a client/server object-based computing system including a first computing device and a second computing device, a data packet representing an object in which the second computing device has interest by placing the packet of data in a queue of objects in which the second computing device has interest using the first computing system, the queue being maintained by the first computing system and arranged to prioritize the packet of data with respect to any other packets of data included in the queue, as claimed by the Applicants in independent claims 1, 9, 14, 18, and 34. Therefore, the Applicants respectfully submit that independent claims 1, 9, 14, 18, and 34 fully satisfy the requirements of 35 U.S.C. §103 and are patentable thereunder.

Dependent claims 3-5, 10, 12-13, 16-17, and 19-20 depend, either directly or indirectly, from claims 1, 9, 14 and 18 and recite additional features. As such, and for at least the same reasons set forth above, the Applicants submit that claims 3-5, 10, 12-13, 16-17, and 19-20 are also not made obvious by the teachings of Lathrop in view of Wesley and further in view of Ma. Therefore, the Applicants submit that dependent claims 3-5, 10, 12-13, 16-17, and 19-20 also fully satisfy the requirements of 35 U.S.C. §103 and are patentable thereunder. Accordingly, the Applicants respectfully request that the rejection of claims 1, 3-6, 9, 10, 12-14, 16-20 under 35 U.S.C. §103 be withdrawn.

III. CONCLUSION

Thus, the Applicants submit that all of the presented claims fully satisfy the requirements of 35 U.S.C. §101 and 35 U.S.C. §103. Consequently, the Applicants believe that all these claims are presently in condition for allowance. Accordingly, both reconsideration of this application and its swift passage to issue are earnestly solicited.

If, however, the Examiner believes that there are any unresolved issues requiring the issuance of a final action in any of the claims now pending in the application, it is requested that the Examiner telephone Mr. Kin-Wah Tong, Esq. at (732) 530-9404 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,

8/21/07

Kin-Wah Tong
Reg. No. 39,400
(732) 530-9404

Patterson & Sheridan, LLP
595 Shrewsbury Avenue
Suite 100
Shrewsbury, NJ 07702